Towards achieving agility in web-based virtual enterprises: a decision-centric approach

Jitesh H. Panchal and Dirk Schaefer*

Systems Realization Laboratory,
The George W. Woodruff School of Mechanical Engineering,
Georgia Institute of Technology,
210 Technology Circle,
Savannah, GA 31407, USA
Fax: +1-912-966-7928
E-mail: jitesh.panchal@me.gatech.edu
E-mail: dirk.schaefer@me.gatech.edu
*Corresponding author

Abstract: Virtual enterprises are groups of loosely connected companies, each providing certain core competencies and working collaboratively towards achieving a common objective. *Agility* of a web-based virtual enterprise refers to its capability to successfully and rapidly adapt to changes in its operating environment. Although, agility has been addressed in the literature with regard to manufacturing processes and product requirements, design processes have not been leveraged to address agility in web-based virtual enterprises. In this paper, we present a strategy that involves (1) designing the design processes along with products, (2) flexibility in the interfaces between different parts of a virtual enterprise, (3) standardisation and (4) integrated modelling of processes associated with all elements of a value chain. We propose a domain independent decision-centric framework for modelling the value chain processes. It allows a variety of stakeholders to structure, organise and model processes using a common framework. It also balances the need for flexibility of interfaces and the standardisation of information from individual stakeholders, thereby providing adaptability to changes in the web-based virtual enterprise.

Keywords: virtual enterprises; agility; decision-centric; templates.

Reference to this paper should be made as follows: Panchal, J.H. and Schaefer, D. (2007) 'Towards achieving agility in web-based virtual enterprises: a decision-centric approach', *Int. J. Internet Manufacturing and Services*, Vol. 1, No. 1, pp.51–74.

Biographical notes: Jitesh H. Panchal is a Visiting Assistant Professor in the Woodruff School of Mechanical Engineering at Georgia Institute of Technology, Savannah. He received his BTech from Indian Institute of Technology at Guwahati and an MS and a PhD in Mechanical Engineering from Georgia Institute of Technology. His research interests are in the field of engineering systems design. Specifically, his current research focus is on simulation-based distributed design methodologies and technologies for multiscale systems. He is a Member of the American Society of Mechanical Engineers (ASME) and the American Institute of Aeronautics and Astronautics (AIAA).

Dirk Schaefer is an Assistant Professor in the George W. Woodruff School of Mechanical Engineering at Georgia Tech Savannah, USA. His research interests are focused on the high-impact interdisciplinary area of information engineering for complex engineered systems. Prior to joining Georgia Tech, he was a Lecturer in the School of Engineering at Durham University, UK. He has published around 50 papers on Computer-Aided Engineering and Design in conference proceedings, journals and books. He is a Member of several professional affiliations including The American Society of Mechanical Engineers (ASME), The Institute of Electrical and Electronics Engineers (IEEE), The Association for Computing Machinery (ACM), The American Society for Engineering Education (ASEE), The Institution of Engineering Designers (IED) and The British Computer Society (BCS). Furthermore, he is a registered professional European Engineer (Eur Ing), a Chartered Engineer (CEng), a Chartered IT-Professional (CITP) and a Fellow of The Higher Education Academy in the UK.

1 Introduction: frame of reference – agility in next generation web-based virtual enterprises

Modern complex engineered systems such as automobiles or passenger airliners are typically created concurrently in networks of highly sophisticated design and manufacture entities. These networks are called *virtual enterprises* and represent a way of organising business activities where different and independent partners exploit business opportunities by establishing an enterprise cooperation (Goldman et al., 1995). They are groups of loosely connected companies, each providing certain core competencies within the virtual enterprise. Most of the current virtual enterprises exchange information over the internet over a web-based framework. These companies are loosely connected because their partnerships exist for limited periods of time. In this regard, key requirements to a successful formation of a web-based virtual enterprise include the process of searching and identifying appropriate partners, and establishing partnerships and mechanisms for information flow.

Appropriate formation of a web-based virtual enterprise is dependent on the products, processes and services to be realised and requires the analysis of each potential partner's core competencies and their coherence with what the web-based virtual enterprise seeks to offer in order to meet market requirements. Seepersad et al. (2002) have shown that market requirements are subject to change, which may have a strong impact on the design and development of a product, the core competencies required, and hence on the formation of the virtual enterprise creating it. Changes in market requirements are associated with significant uncertainty. Specifically, it is hard to predict, if at all, what changes are going to occur and hence how to respond to them. In certain scenarios, the same set of partners can fulfil the changing needs whereas, in other scenarios, new partners need to be included as a part of the web-based virtual enterprise, thereby changing the product realisation process. A virtual enterprise's capability to successfully and rapidly adapt to changes is referred to as *agility* and is one of the most important factors for sustaining competitiveness in turbulent markets. In the rest of this paper, we present the essential aspects of agility in a virtual design and manufacturing enterprise.

2 Agile enterprises: existing efforts and research challenges

2.1 Existing efforts on agile design and manufacturing

The concepts of design and agility are widely discussed in the literature. Design is the process of transforming a set of potentially ill-defined customer requirements into a physical artefact. This process is a combination of determining appropriate attributes and then determining their values. A number of methodologies exist for deployment in different application areas and different corporate cultures. Agility is seen to be the ability to rapidly react to changes in the environment, be they expected or not. When applied to manufacture, this is seen to be the ability to change a manufacture resource rapidly to produce a different assembly or product. Thus, an agile manufacturer is able to rapidly respond to changes in product demand. Agile manufacturing is a well established and researched methodology which has grown from lean manufacturing. Goldman et al. (1995) consider agility as a comprehensive response to the challenges posed by a business environment dominated by change and uncertainty. In other words, one underpinning principle of agility is responsiveness to a changing business environment. According to Maropoulos et al. (2004), there are three routes a definition of agile design could take from a design-centric perspective: firstly, there is design for agile manufacturing, ensuring that a product is designed with the manufacturing capabilities at the forefront of the design constraints. In this way, a product can be manufactured with the least effort, time and cost (Corbett et al., 1991). Secondly, one may define agile design as any product which has agile functionality. This concept of a product which once manufactured can have multiple functions, or even have its function changed entirely, has previously been defined as adaptable design by Gu et al. (2004) who propose a modular design with functionally independent components or subassemblies. The idea of adaptability to changes has also been referred to as open engineering systems by Simpson et al. (1998). With simple interfaces between modules, upgrades are more feasible and therefore more likely. The concept of modularity is also proposed by Suh (1990) as the axiomatic design independence axiom. Finally, Maropoulos et al. (2004) define agile design as "a flexible, scaleable, adaptive, responsive design process, encompassing not just design, but all processes which impact upon that design, from definition of requirements through to end-of-life".

The general need for agile response in dynamic environments has also been analysed in the context of manufacturing (Jiang and Fung, 2003; Lau et al., 2003). However, agility with respect to engineering design has only recently been addressed by Matthews et al. (2006). It is a new research aspect that is likely to have a strong impact on the design of open complex engineered systems in the future (Simpson et al., 1998). Systematic linear design approaches, such as Pahl and Beitz (1996), are characterised by a series of gates that determine if the current quality of the design is sufficient to proceed to the next phase. They provide a strong basic design process, which is often used within Concurrent Engineering (CE) frameworks. CE frameworks distribute independent design tasks to design agents that work concurrently and thus significantly reduce total development time (Kusar et al., 2004). However, traditional design approaches lack the ability to respond to unpredictable changes. With regard to this, and taking into account the definition suggested by Maropoulos et al. (2004), agility may be seen as the ability to rapidly react to changes in the environment, that is, to predictable or unpredictable events. In this context, responsiveness to unpredictable events is seen to be the major distinction between agile design and concurrent design. Given a design brief, both methods allocate design tasks to a pool of design agents. When an unpredicted event occurs, which could be, for example, a late customer request, the failure of an agent or some other external environmental impact, the concurrent design process is interrupted. However, using an agile design methodology would allow evaluating the impact of the event and *restructuring the design process accordingly* with the aim to minimise the total impact to the design process. Matthews et al. (2006) introduce a framework for representing and analysing an agile design methodology based on CE. They illustrate how the design state could be mapped out onto the web-based virtual enterprise network, and provide a means for representing and measuring the effect of unexpected events that may occur during the design process. They also propose a classification scheme for the severity of unexpected events and their agile design methodology redistributes the design work after an event occurs according to the degree of severity. Their proposed framework enables web-based virtual enterprises to test various work redistribution policies by using statistical methods to determine the most suitable one for different types of events.

As a summary, the existing efforts on agility are primarily focused on the product, and associated manufacturing processes. An important aspect that has not been addressed is the *design of the design process* itself to maximise agility. Design processes determine to a large extent the efficiency and effectiveness with which design objectives are accomplished (Simon, 1996). Further, design process strategies can affect not only the efficiency with which the resources for designing are used, but the nature of the virtual enterprise as well. Hence, in order to achieve agility in enterprises, it becomes very important to design the design processes along with the products. The purpose of this paper is twofold. Firstly, the concepts of agility and design are brought together in the context of virtual enterprises. Secondly, a new approach to integrating the design of a product with the design of the design process itself is proposed. This is a key novelty, which bears the potential to significantly enhance the degree of agility that can be achieved within future web-based virtual enterprises.

2.2 Requirements for agility in web-based virtual enterprises

The agility of web-based virtual enterprises is measured by their ability to adapt to changes in the marketplace. These can be attributed to changes in customer requirements, demand or supply of essential raw material, availability of new technologies or anything else. In order to adapt to such changes quickly, we believe that there are at least four key requirements (see Table 1). The first key requirement is the 'consideration of design processes in establishing the virtual enterprises'. This requirement is important because in addition to the manufacturing capabilities, it is important to include the core design competencies during the development of a web-based virtual enterprise. In order to address this requirement, we propose the design of design processes along with the products while considering the need for adaptability at the same time. The design of design processes would lead to the development of a part of the web-based virtual enterprise that is suited for specific products to be designed and manufactured. The second related key requirement is the *capability to restructure the design processes* in case of changes. The proposed strategy to address this requirement is to develop flexible interfaces between different parts of a web-based virtual enterprise. The third requirement is *interoperability between various entities in a virtual enterprise*. Standardisation is proposed as a strategy to address the requirement of interoperability between various entities.

Table 1 Requirements for agility in web-based virtual enterprises and strategies to address these requirements

	Requirements for agility in virtual enterprises	*Proposed strategy*
1	Consideration of design processes in establishing virtual enterprises	Designing design processes along with products
2	Capability to restructure design processes	Flexible interfaces between different parts of a virtual enterprise
3	Interoperability between various entities in a virtual enterprise	Standardisation
4	Integration and management of information throughout the life cycle of a virtual enterprise	Integrated modelling of processes associated with all elements of a value chain

The fourth requirement is the 'integration and management of information throughout the life cycle of the virtual enterprise'. This is particularly important because of the dynamic nature of the enterprise where partnerships are established for relatively short periods of time. Further, it is important to integrate the information captured by different entities in different Product Data Management (PDM), Product Life cycle Management (PLM), Enterprise Resource Planning (ERP) systems, etc. The proposed strategy to address this requirement involves integrated modelling of processes associated with all elements of a value chain. These elements of the proposed strategy are highlighted in Section 3 and discussed in detail in Section 4.

3 Elements of a strategy for addressing the requirements for agile web-based virtual enterprises

In this section, we outline the enabling strategies that, we believe, would facilitate the web-based virtual enterprise that dynamically form coalitions on a need basis. This essentially means that the components of a virtual enterprise are completely modular and can be combined together in different configurations. As discussed in Section 2.2, we believe that the key facilitators are:

1 designing design processes along with products

2 flexible interfaces between different parts of a virtual enterprise

3 standardisation and

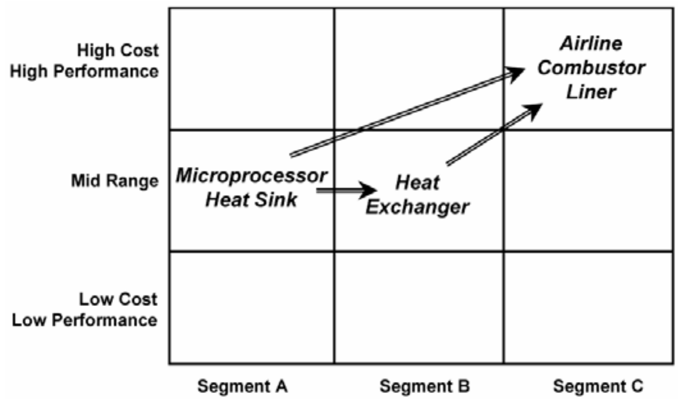
4 integrated modelling of processes associated with all elements of a value chain.

Each of these facilitators is discussed in Sections 3.1–3.4, starting with a motivating scenario of a fictitious company – LCADes.

3.1 Designing design processes along with products

Imagine a materials design and manufacturing company named LCADes that designs materials called Linear Cellular Alloys (LCAs) (Cochran et al., 2000) at a mesoscopic level. These LCAs are cellular materials, have extended prismatic cells and are manufactured through extrusion of slurry of metallic powders, subjected to drying (Hayes et al., 2001). These materials are suitable for multifunctional applications that involve not only good structural properties but also good thermal properties. These LCAs can be designed to obtain desired properties. Currently, LCADes designs and manufactures LCAs for microprocessor heat sinks. LCADes is capable of quickly designing heat sinks for microprocessors with different heat generation rates. Now imagine that LCADes wishes to extend the current design operations to design of LCAs for heat exchangers and for combustor liners. This is shown on the market segmentation grid in Figure 1. It is the goal of LCADes to move from the microprocessor segment to heat exchanger and combustor liner segment as fast as possible.

Figure 1 Leveraging design process knowledge for satisfying new demands



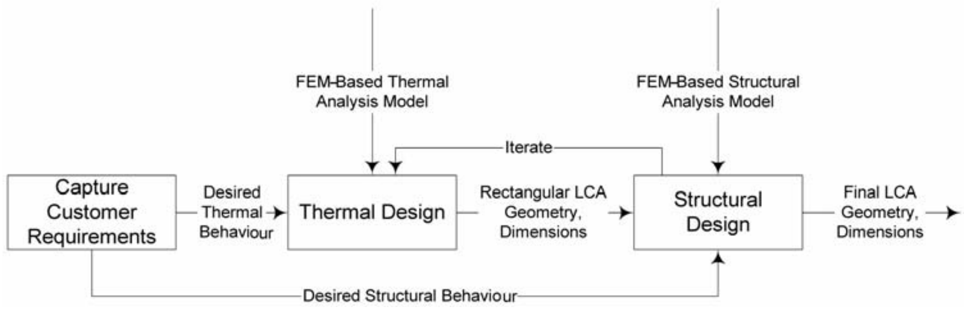
As suggested by Meyer (1997), this shift across various market segments can be achieved through development of product platforms and leveraging. However in this case, from the standpoint of designers, there are no product components that can be directly reused. Hence, product architectures cannot be used for this design scenario. Similar situation of inability to leverage product components is faced with other material design scenarios where the design is carried out at a microstructure scale. Hence, the question that LCADes faces is: How can LCADes leverage design knowledge so as to facilitate faster development of new products? This question can be answered by recognising that 'an important component of the design knowledge in a company like LCADes is the design process'. Our belief is that if LCADes *designs the design processes* along with the products in an open fashion, it would be possible to leverage design knowledge. Some examples of design process knowledge that can be leveraged across design of heat sinks, heat exchangers and combustor liners are design considerations, design methods and tools.

Let us consider various design process options for designing a LCA requiring high heat transfer rate and high stiffness. The design variables considered are: cell shape, total height of the LCA, thickness of the cell walls and fluid velocity. The process of designing LCAs involves various steps such as cell shape selection, structural analysis, thermal analysis, design space exploration, geometry refinement, etc. Assume that the designers are restricted to use either triangular or rectangular cells. The process can be structured quite differently depending on the designers' specific needs. Some example scenarios of LCA design processes are discussed next.

3.1.1. Scenario 1: sequential design, thermal first

In this scenario (see Figure 2), the thermal designer fixes some variables in the design space and passes on the design to the structural designer. Given the choice between rectangular cell shape and triangular cell shape, the thermal designer chooses a rectangular cell shape because of superior forced conjugate (conduction and convection) heat transfer performance.

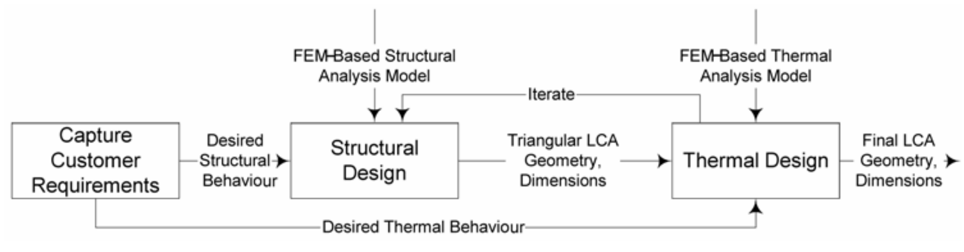
Figure 2 LCA design scenario 1: thermal first sequential design



3.1.2 Scenario 2: sequential design, structural first

In this scenario (see Figure 3), the structural goal is more important than the thermal goal. The structural designer determines the structure and then passes the resulting geometry to the thermal designer for modification. In this case, the structural designer selects triangular cells in lieu of less stiff rectangular cells.

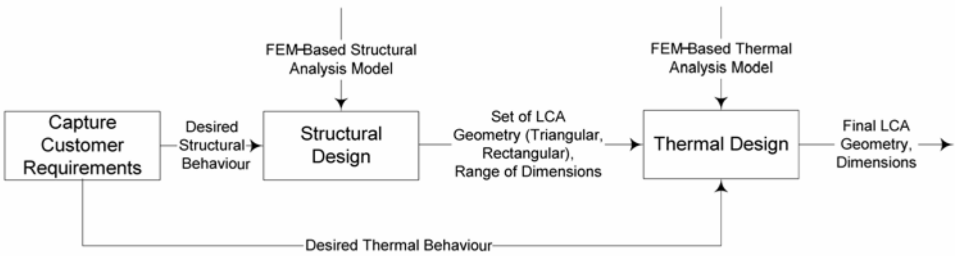
Figure 3 LCA design scenario 2: structural first sequential design



3.1.3 Scenario 3: set-based design

In the set-based design scenario (see Figure 4), designers consider sets of design alternatives rather than pursuing one alternative directly. The philosophy is to gradually narrow down the design space until a final solution is achieved. In the LCA design scenario, this may be implemented as one designer (thermal or structural) synthesising a range of design parameters and then passing on this range to another designer to select the best value in that range. Since the designers do not pick a single alternative, the designers develop both cell topologies – triangular and rectangular. Although, this approach is more likely to result in designs that show superior performance with regard to both thermal and structural considerations, the design effort involved in developing all alternatives is higher.

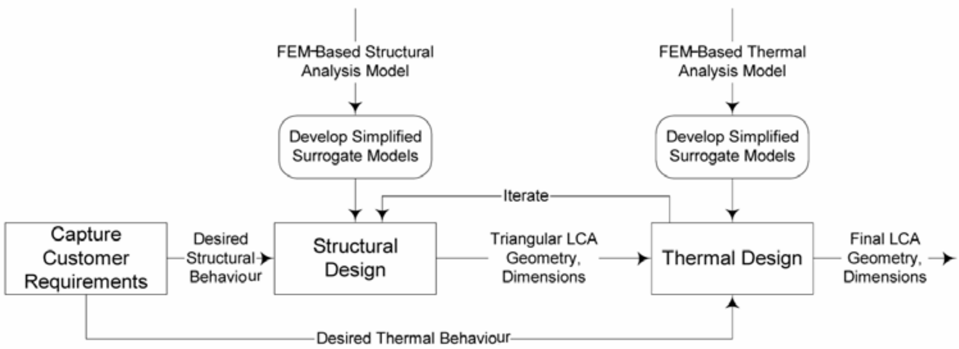
Figure 4 LCA design scenario 3: set-based design



3.1.4 Scenario 4: use of surrogate models

The computational intensity of analysis models associated with design is often substantial. In these instances, it becomes necessary to develop surrogate models to replace expensive computational runs. These surrogate models, however, are not exact and may introduce additional error. In the LCA design example (see Figure 5), simple response surface models can replace computationally intensive FEM analysis codes. The choice of appropriate models also depends upon progress made along the design process. In the earlier stages, it is not possible to use high fidelity analysis models because of limited knowledge regarding the design. However, in the latter stages of design, when the design specifications have been determined, high fidelity analysis models are usually more appropriate.

Figure 5 LCA design scenario 4: use of surrogate models

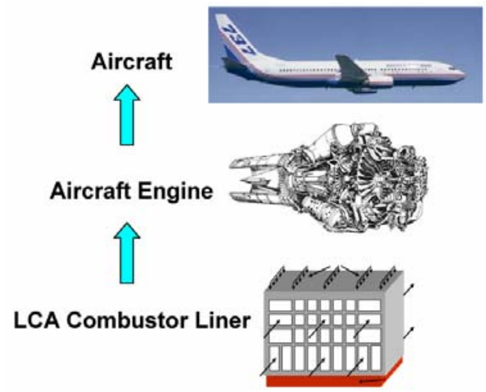


3.1.5 Scenario 5: parallel iterative design

Another design process option can involve performing the design activities in parallel. In a parallel iterative design process for multifunctional applications, concurrent, point-based analysis is carried out for structural and thermal requirements. These analyses provide information about the simulated behaviour for a given loading (both thermal and structural). This simulated behaviour is compared to expected behaviour. If these do not match, appropriate changes must be made to the geometric parameters to obtain the desired performance. The process is continued until the designers converge on a mutually acceptable solution.

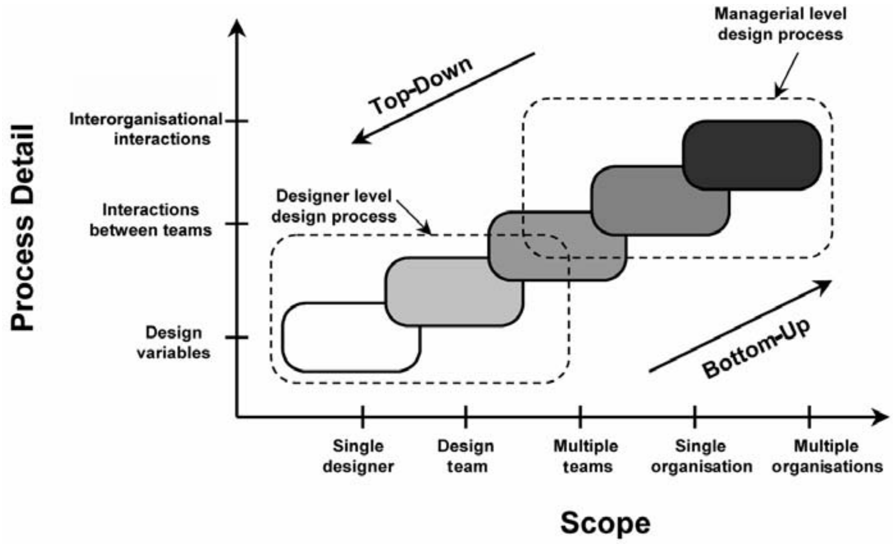
Until this point in this section, we have discussed the different design process strategies for designing LCAs with fixed requirements. Taking a step further, let us imagine a design chain where LCADes performs design for another design enterprise, which in turn provides information to a third enterprise (see Figure 6). LCA-based combustor liner is only a subcomponent of a bigger system like an aircraft engine, which can be designed by second enterprise. The aircraft engine is a subcomponent of a bigger system – the aircraft, which can be designed by third enterprise. Hence, 'such complex products are designed by interacting design teams in a hierarchical fashion'. Associated with this product hierarchy, there is also a hierarchy of design processes. In the context of complete aircraft design, process can be designed at multiple-organisations level. The aircraft engine can be designed at multiple-teams level and the process of design of LCA can be designed at a single team level. As shown in Figure 7, design processes can be designed at various levels based on the scope and the detail. As the scope of design process increases from single designer through teams leading to multiple organisations, the detail of design process changes from design variables level to modelling interorganisational interactions. In other words, we can design the design process in a top-down manner if we are performing original design of an entirely new product. In the case of original design, the question faced by design firms is: 'How can the firms capture top-down design processes for reuse?' If the firm is involved in adaptive or variant design, the question is: 'How can design firms leverage design processes from lower level design processes to facilitate bottom up design of design processes?'

Figure 6 A component of design chain for aircraft design



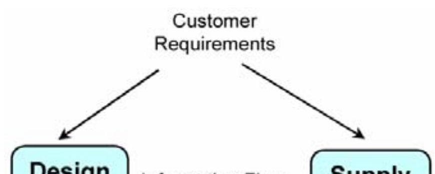
Product design processes are increasingly becoming like a supply chain where instead material, information flows between different levels of designers. The configuration of these design-chains and the manner of interactions between designers affects to a great extent the efficiency and effectiveness of design processes. Design tasks are increasingly outsourced, thereby posing a problem of managing design chains. Diverse customer requirements and mass customisation of products also pose a challenge of quick reconfiguration of design chains. This calls for 'designing not only the products but also processes in design chains'.

Figure 7 Design processes at different levels of details and scope



The need for rapid translation of business ideas into products satisfying individual customer requirements necessitates changes in both the process of design, manufacturing and the supply chain partnerships. In Figure 8, the information-based interactions between design chains and supply chains are shown. Changes in the customer requirements may require costly changes in both these chains. This necessitates companies to perform design in a strategic manner. Strategic design is a comprehensive approach for forecasting changes or shifts in markets, associated customer requirements and technological capabilities and for devising artefacts that accommodate these shifts (Seepersad et al., 2002). One of the key components of strategic design is designing design processes and associated design chains. The objective is to design the design chains in such a manner that it is robust to changes in external factors. In order to address this challenge, there is a need to develop methods grounded in engineering design theory for modelling and design of design processes. This would also enable integration of design chains and supply chains. As an enabler to designing design processes, there is a need to model the processes and design chains in a reusable manner.

Figure 8 Interactions between design chains and supply chains



An important aspect of modelling value chain processes is the ability to model the interactions between stakeholders in a flexible manner. Flexibility in these interfaces has the potential to allow plug-and-play between different components of the enterprise. The idea of flexible interfaces is discussed next.

3.2 Flexible interfaces between different parts of a virtual enterprise

Modularity is an essential feature of a future web-based virtual enterprise. Modularity refers to the independence of various components of the system, the components being integrated together with well-defined interfaces. Much attention has been given recently to a modular approach to product design. 'Plug-and-play' is a single most important aspect of some areas like software engineering. In the context of web-based virtual enterprises, we believe that more attention needs to be given to *organisational and process level interfaces*. *Organisational level interfaces* refer to the interfaces between various organisations collaborating in a distributed environment. *Process level interfaces* refer to the interaction between various groups in the company. In the global enterprise, multiple organisations will collaborate to realise a common goal. These organisations have very different organisational set-ups, work cultures, ethics, time zones, cultures, languages, etc. In such a heterogeneous system, effectiveness of collaboration is an important issue. One possible way to achieve this is to homogenise the system, that is, to adopt some common standards of working styles. For example, there can be a common code of operation for the companies participating in the collaboration. Either the companies can jointly define these common standards or there can be a central governing authority that is responsible for setting these standards. However, we feel that it is difficult to achieve a homogenisation of cultures and ethics. Hence, there needs to be some interface that maps the working environment of one organisation to that of the other. All collaborating organisations can maintain their own working philosophy. Some communication ports can be set up between the collaborating organisations. All the exchange of knowledge can take place through these communication ports. A simple example of these communication ports can be that of a translator who operates between two businesspeople speaking different languages. The translator understands both the languages and can transfer ideas between these two people. This is an example of simplest type of interface required. The situation becomes very complex when we consider developing communication ports for issues like ethics, culture, values, etc. Other issues while designing organisational level interfaces are: mapping mismatched priorities, asynchronous sharing of critical information and loss of knowledge between organisations.

To address this challenge of flexible interfaces between organisations, it is important to start with modelling interactions among design process elements. Stakeholder collaboration patterns need to be regulated, coordinated, standardised and captured in order to ensure conciseness and promote efficiency. The proposed approach is to facilitate creation of design chains, increase modularity by isolating effects make it possible to more clearly evaluate the value proposition of potential stakeholders contributing to the product development effort. In the following section, we present our thoughts on the need for a common framework for managing value chain processes, including design and supply chain processes.

3.3 Standardisation

As we have discussed previously, heterogeneity is a key characteristic of a web-based virtual enterprise. Standardisation is a means of achieving interoperability between heterogeneous components of the enterprise. Standardisation provides flexibility and increases reuse. As shown in Figure 9, standardisation can be carried out for various modules in a product. Standardisation can be carried out for various interfaces between the modules. And the design process itself can be standardised. The levels at which standardisation can be done are: between modules, across the product family or across organisations. Another dimension to standardisation is the level of standardisation (see Figure 10). These levels of standardisation are:

1 between components in a product
2 across the product family and
3 across various organisations.

The level of standardisation in a company depends on the business requirements. We believe that the organisational level of standardisation is very important and needs to be taken into consideration right from the initial stages in product development cycle.

Figure 9 Standardisation can be done for modules, interfaces and process

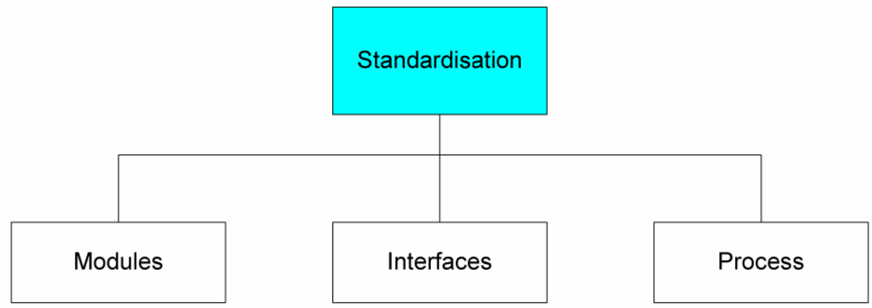
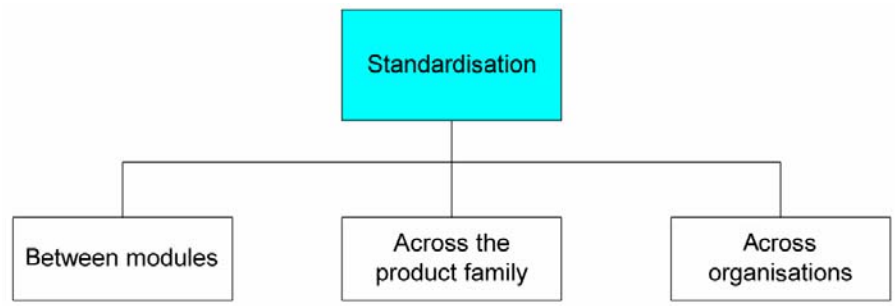


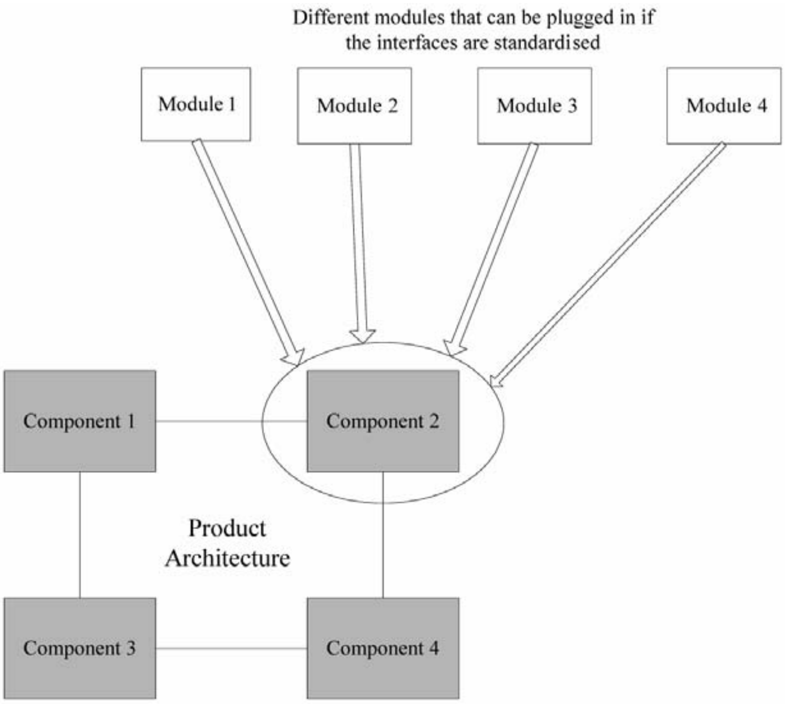
Figure 10 Levels of standardisation – among modules, across the product family and across organisations



The philosophy of mass customisation is to produce products with varied functionality while maintaining minimal changes to the product itself. The base product architecture remains the same for different products in the product family. Having standardised components helps in reducing production and procurement costs. In a mass customised product, the components can be mass-produced if they are standardised. This provides the benefits of economies of scale.

Standardising interfaces between various components provides the capability of easy reconfiguration. If the interfaces between modules are standardised, these components can be easily replaced by different components (see Figure 11). A base architecture can be used for designing a product family where the modules are interchangeable.

Figure 11 Different modules can be inserted in product architecture if the interfaces are standardised



Standardisation of process relates to using common design-manufacturing processes for different products in a product family. To some extent, using standardised modules and interfaces aids in the standardisation of a process. Standardised process reduces this set up time. Standardisation helps in introducing benefits of mass production in a mass customised product line. Standardisation of process components allows us to capture company's best practices as patterns. If these processes are computer executable, they can be captured in the form of executable templates.

3.4 Integrated modelling of processes associated with all elements of a value chain

Due to increasing globalisation and outsourcing, the infrastructure supporting design activities is likely to become just as federated as those underlying the already vast network of suppliers involved in any product realisation effort. Due to this impending reality, there is a need for the integrated modelling of processes associated with all the elements of value chains. Two essential aspects of a value chain include design chains and supply chains. Currently, there are independent models for both the supply chains and design chains. For example, the SCOR model (Supply Chain Council, 2004) is developed to represent and measure supply chains in a standardised manner to enable improvements in supply chain operations through analysis of current processes and best practice emulation. Along these lines, numerous case studies have been conducted. The SCOR model is extended to the Enterprise Transaction Model by Streamline SCM (2004).

Similarly, there are separate models for representing design processes at different levels. The intent behind modelling processes at the *strategic design level*, as discussed earlier, is mainly that of identifying needs and ascertaining the technological capability of meeting them. The formulation of the required strategy is aided by the formalised identification and subsequent evaluation of both in-house and external resources – resulting in a suitable value chain. At the *designer level*, on the other hand, the goal is to manage information flows, integrate models, tools and facilitate decision -making. Both ends of the spectrum are crucial for modelling and integrating the design chain into the overarching value chain. So far, both ends are addressed independently, in an isolated fashion. This is true whether referring to a single original design or an adaptive, a variant or a derivative design resulting there from.

To address the challenge of integrating domain dependent process models, we propose a *generic, decision-centric framework* for enabling design chain integration with supply chains. This model would also facilitate integration of models at different abstraction levels, such as strategic design level and designer levels, as discussed above. The domain independence of the underlying constructs makes them amenable to integration with other value chain process models like SCOR (Supply Chain Council, 2004) in order to address the future needs of federated enterprises. Primary benefits of adopting this perspective include domain independence and flexibility. Other benefits are ease of identifying relationships and conciseness of isolated perspectives and comprehensive way of structuring design processes. Decisions are the natural points for resource commitment, deliberate reduction of design freedom and determination of form, function and behaviour. With this in mind, decisions serve as milestones in the design processes and essentially constitute the nexus for collaboration. The decision-centric framework is presented in Section 4.

4 Proposed decision-centric approach for agile enterprises

One of the primary reasons for the incapability of current web-based frameworks to model generic design processes at various levels is that they are developed based on a *tool-centric* view of design processes. According to a tool-centric view, a design process is a network comprised of software tools employed for processing information. The adoption of a tool-centric perspective in developing design frameworks thus focuses the effort on *achieving interoperability* between

1 different tools that perform similar functions (such as different CAD applications)

2 tools providing different functionality (structural analysis, crash, vibration, etc.) and

3 applications pertaining to different domains.

Interoperability between such tools is achieved by various standards such as STEP, XML and UML. Recently, Peak et al. (2004) proposed a *model-centric* perspective to support the further development of these frameworks. Specifically, a product information model is the central core which is populated using relevant tools. Such a model-centric view constitutes a significant improvement over the tool-centric view, commonly espoused, because information is no longer tied solely to the particular tools used for its creation or modification. We acknowledge that a model-centric perspective is important for realising the seamless integration of information models, associated with different aspects of product design and useful for guiding the development of CAE and PLM frameworks to support fine grained interoperability, as well as the development of a collective product model. However, we assert that neither the tool- nor model-centric perspectives (alone or in concert) are adequate for effectively supporting the reuse of design processes due to their 'inability to capture the problem definition independent of the solution procedure'. In fact, such tools are primarily used to capture procedural aspects. Put another way, current tools do not capture

1 *what* the design problem is

2 *how* the designer partitions the problem and

3 *how* different problems are related.

Instead, current tools only capture the specific series of steps a designer adopts when solving the problem at hand in a quasi documentary fashion. Design problem changes resulting from the changes in the product to be designed, its design variables, constraints, analysis codes, etc. can thus not be translated to the procedural information captured within the individual tools. The word 'problem' has been used in many different ways in the engineering design community. In this paper, we define a problem as 'either an obstacle to be overcome or a question to be answered' (Muster and Mistree, 1988). This definition is different from the textbook type problem-solving, where the problem is completely defined and can be solved using a predefined set of steps resulting in a unique solution (Hazelrigg, 1998). Without capturing the problem definition and the solution procedure independently of each other, it is difficult to support reuse of design processes in the web-based frameworks.

4.1 Decision-centric modelling of processes

The decision-centric view of design addresses the limitations of model- and tool-centric views. From a decision-centric perspective, the emphasis is on modelling design processes as a network of decisions. According to many researchers such as Hazelrigg (1998), Muster and Mistree (1988) and Thurston (1999) the fundamental premise of decision-based design is that engineering design is primarily a decision-making process. Specific advantages of adopting a decision-centric perspective include the ease with which both model- and tool-centric views are generated. Furthermore, domain independent representation of design processes becomes feasible. Hazelrigg describes decision-based design as *omni-disciplinary,* "the seed that glues together the heretofore disparate engineering disciplines as well as economics, marketing, business, operations research, probability theory, optimization and others" (Hazelrigg, 1998). Herrmann and Schmidt (2002) describe a complete product development organisation as a network of decision-makers who use and create information to develop a product. Although, principles of decision-based design have been accepted in theoretical aspects of design research, they have not been implemented in design frameworks. Current tools do not capture information related to designers' decisions; decision related information is captured in the form of meta-data (if at all). In this paper, we use decision-centric design to model the building blocks of design processes in the form of decision problems.

The underlying need for reusability of design process related information necessitates representing information in a domain neutral form that supports designers in providing and structuring required information content in a reusable fashion. This in turn calls for a domain independent means of capturing design information. In order to facilitate designer interactions, required for effective collaboration from a decision-based perspective, expression of information related to design decisions in a standardised format is also required. Such a standardised form for representing information is provided by the Decision Support Problem (DSP) Technique proposed by Bascaran et al. (1989), Fernández et al. (2005), Mistree et al. (1989, 1993) and Muster and Mistree (1988), specifically the compromise Decision Support Problem (cDSP) (Mistree et al., 1992). In the DSP Technique, support for human judgement in designing is offered through the formulation and solution of DSPs, which provide a means for modelling decisions encountered in design. As an example, the formulation of a cDSP is shown in Figure 12. The cDSP formulation consists of four key sections

1 given
2 find
3 satisfy and
4 minimise.

In the 'given' section the information available to designers for decision making, which includes the available simulation models that generate information about the system's behaviour and a designers' preferences, is captured. In the 'find' section of the cDSP, information about the design variables that designers can control in order to satisfy the design objectives is captured. The information about bounds on design variables, any problem constraints, and the design goals is captured in the 'satisfy' section of the cDSP. The overall objective function to be minimised is captured in the 'Minimise' section. A similar formulation has been developed for selection decisions (Bascaran et al., 1989) and its variant – utility-based selection (Fernández et al., 2005).

Figure 12 The cDSP – mathematical formulation

Given

An alternative to be improved

Target value for goals, G_i, $i = 1,\ldots,n$

Relative importances of goals, w_i, $i = 1,\ldots,n$

Find

System Variables: X_j, $j = 1,\ldots,m$

Deviation Variables: d_i^-, d_i^+, $i = 1,\ldots,n$

Satisfy

Goals: $A_i(X) + d_i^- - d_i^+ = G_i$, $i = 1,\ldots,n$

Constraints: $C_k(X) \leq 0$, $k = 1,\ldots,p$

$d_i^- \bullet d_i^+ = 0$, $d_i^-, d_i^+ \geq 0$

Bounds: $lb_j \leq X_j \leq ub_j$, $j = 1,\ldots,m$

Minimise

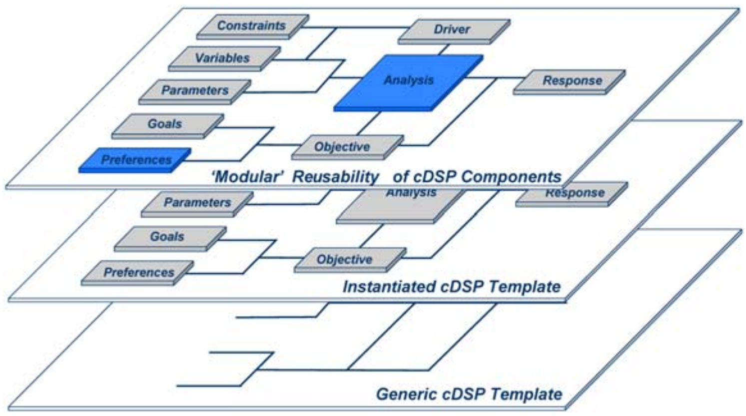
Deviation Function:

$Z = \Sigma w_i \bullet d_i^+ + \Sigma w_i \bullet d_i^-$, $i = 1,\ldots,n$

These DSPs are embodied in the form of computational templates as shown in Figure 13. All the aspects of cDSP including variables, parameters, goals, etc. are shown in the figure. In addition to this, the *driver* refers to the optimisation algorithm to be used for minimising the objective function. The templates resulting from the mathematically modelled decisions are defined as computer-based representations of DSPs. These DSPs are represented as declarative templates that can be parsed, stored and reused. Since these keywords and descriptors are domain independent, they represent a common structure (or a conceptual schema) for DSPs from any domain. This is one of the most important characteristics of the DSP Technique that *enables reuse of design information across domains*. As a summary, decision-based design is chosen as a framework in this paper because of its

1. domain independence (decisions are common across different engineering domains)
2. design phase independence (the structure of decisions remain the same during any phase of the design process) and
3. ability to be used for modelling processes at other levels modelled using decisions.

Figure 13 Template for compromise DSP



Example application of decision-centric templates for design processes: an example of the application of decision-centric templates for reusing design information across multiple domains is presented by Panchal et al. (2004). In this paper, two different design problems are chosen:

1. design of springs and
2. design of pressure vessels.

While both of these products are described in terms of different geometric constraints, describing their form and mechanical relations describing their function, their design process is similar at the level of design decisions. The information related to the decisions associated with these two problems can be divided into two categories – problem specific (declarative information) and process specific (procedural information). The authors have shown that there is commonality in the procedural information and difference in the declarative information. As an example, although the specific constraints are different, the process of evaluation of constraints and their utilisation in the decision-making process is the same. Other aspects of the process include formulation of goals, designers' preferences, analyses and response evaluation. The authors separate the product specific information that is shown in Figure 14 from the common process information, which is shown in Figure 15. Both the product information and the process information are stored in the form of templates that are computer interpretable, allowing for the execution, reuse/reconfiguration and documentation of design processes and any of their components, respectively. The details of the decision templates are provided in the following section.

Figure 14 Product information for pressure vessel and spring design

cDSP "Chips"	Pressure Vessel		Spring	
Constraints	Stress: $\frac{PR}{T} - S_t \le 0$ Thickness: $5T - R \le 0$ Radius: $R + T - 40 \le 0$ Length: $L + 2R + 2T - 150 \le 0$		Minimum Deflection: $\delta = \frac{8FD^3N}{d^4G} \ge 1.1$ Maximum Solid Height: $H = N \cdot d \le 0.5$	
Variables	Radius, Length, Thickness		Number of Coils, Wire Diameter	
Parameters	Density, Strength, Pressure		Applied Force, Coil Diameter, Shear Modulus	
Goals	Volume Target = 500000 m^3 Weight Target = 300 kg		Stiffness Target = lbf/in Volume Target = in^3	
Preferences	Volume Weighting Factor = 0.5 Weight Weighting Factor = 0.5		Stiffness Weighting Factor = 0.5 Volume Weighting Factor = 0.5	
Objective	**Maximise** Volume $V(R,L) = \pi\left[\frac{4}{3}R^3 + R^2L\right]$ **Minimise** Weight $W(R,T,L) = \pi\rho\left[\frac{4}{3}(R+T)^3 + (R+T)^2L - \left(\frac{4}{3}R^3 + R^2L\right)\right]$		**Maximise** Stiffness $k = \frac{d^4G}{8D^3N}$ **Minimise** Volume $V = \frac{1}{4}\pi^2Dd^2(N+2)$	
Analysis	**Inputs**	**Outputs**	**Inputs**	**Outputs**
	Radius, Length, Thickness, Density, Strength, Pressure	Volume, Weight	Wire Diameter, Coil Diameter, Shear Modulus, Number of Coils	Stiffness, Volume
Driver	**Optimisation Algorithm** – Exhaustive Search, SQP, etc.		**Optimisation Algorithm** – Exhaustive Search, SQP, etc.	
Response	**Design Variable Values** – Radius, Length, Thickness **Objective Function Value** - *Z*		**Design Variable Values** – Radius, Length, Thickness **Objective Function Value** - *Z*	

4.2 *Template-based modelling of decision-centric processes*

The templates for design decisions are based on the compromise DSP construct presented in Section 4.1. The key components of decision templates include information about design variables, responses, parameters, constraints, goals, preferences and objectives. An information model for DSP is shown in Figure 16. The topmost entity is a *DecisionProblem*. This decision problem contains all the declarative information related to a DSP. The decision problem consists of four important elements – design space, response space, problem constraints and preferences. Design space is defined by all the design variables that can be controlled by designers. Design variables can be either real or discrete. Real design variables have a continuous range of values they can assume. Response space is defined by all the parameters that constitute the behaviour space. Parameters in the response space have targets associated with them. These targets are based on the mapping between customer requirements and engineering specifications. Both the design variables and response variables are special types of attributes described in the schema for product model presented in Panchal et al. (2005). It is important to note that the relationship between design variables and response variables is not defined in the problem description, but is defined in the product specific information model. This separation of information is important for reusability.

Figure 15 Process map for design of spring/pressure vessel as modelled in ModelCenter®

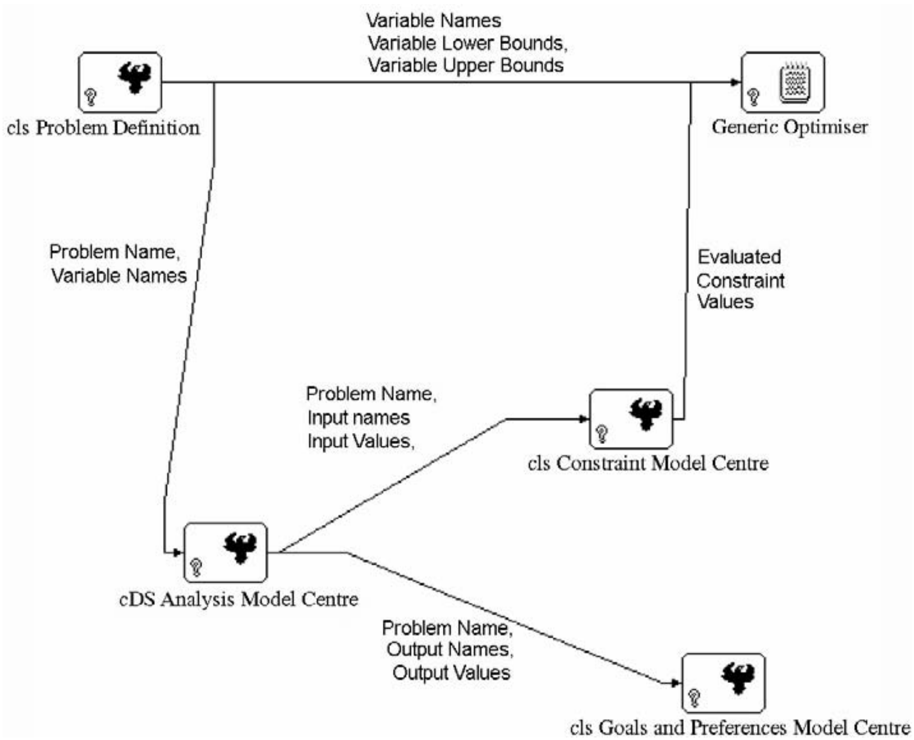
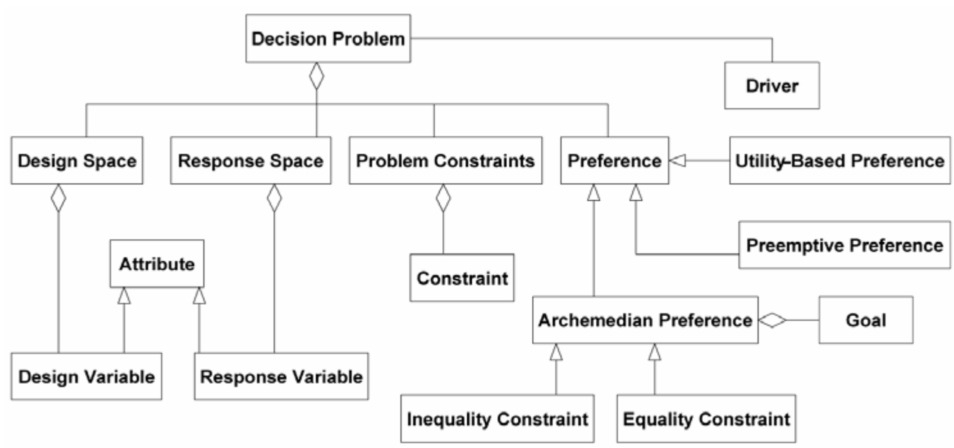


Figure 16 Information model for decision problems



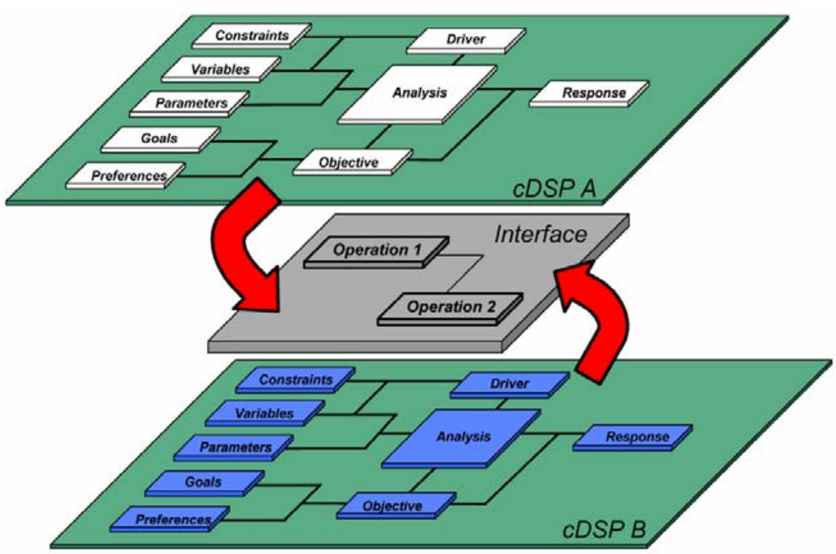
The third element of the design problem is that of problem constraints, the first two being design space and response space. The constraint component of the problem definition captures only the constraints that are due to the manner in which the problem is defined. Product specific constraints are not defined in this section. They are captured using the relationship part of the product model. Constraints can be of two types – equality and inequality. The fourth component of the design problem representation is preference. The preference part of the information model captures how much a designer values different outcomes in a manner that can be mathematically evaluated. These preferences can be captured in different mathematical forms – Archimedean, preemptive or using utility functions. In the Archimedean formulation, different goals are assigned weights and the overall objective function value is evaluated by taking the weighted sum of individual goals. In the preemptive formulation, different levels of objective functions are defined. After the higher levels are satisfied, designers can proceed to satisfy the next level of objective function. Using the utility-based preference representation, the preference values can be defined to vary with the value of each goal. Since multiple goals can be defined, the information model supports multiobjective decision making. In addition to these entities, there is a driver that captures the information about the optimisation algorithm used for executing the decision problem. The information model presented in this section can be extended to selection decisions and can be used as a basis for creating ontology for decision-centric design processes.

Interface templates serve as domain-independent communications protocols for regulating the way in which experts (operating in different functional domains) share information for effective collaboration. An interface in a design process separates or partitions multiple dependent or interdependent designers and their respective design activities. As shown in Figure 17, interface templates serve as means for connecting decision templates to one another in a computer interpretable manner. The nature of the collaboration between designers determines the form of the interface. Appropriate interface templates are developed based upon the underlying informational dependencies between the decision-makers. Consequently, the interactions between decision-makers can be easily adapted by changing the interface template, while reusing the same problem formulations. Since the decision templates and their instantiation remains the same, the designers still control the formulation of their own decisions. The required level of modularity is maintained via the development of domain independent interface templates that are distinct from the decision templates being linked. The notion of linking various decision templates via interface templates is illustrated in Figure 17, where the decisions corresponding to two design subproblems are instantiated as distinct cDSP templates.

Different types of interface templates can be depending on the type of interaction between decisions. For example, game theoretic protocols (Marston, 2000; Xiao et al., 2002) can be used to model cooperation, non-cooperation and leader follower scenarios of decision interactions. More details about the implementation of these protocols is presented in Panchal et al. (2004) and Schnell et al. (2006).

An example of utilisation of interface templates is presented by Schnell et al. (2006). The authors utilise the interface templates to model the interactions between designers and manufacturers in a modular fashion. Game-theory-based protocols are used to mathematically model the information flow between compromise decisions associated with design and manufacturing. The interfaces can be changed while keeping the same decision formulations because the information contained in the interface template is independent of the information in the decision templates. Hence, by using the interface templates, different kinds of collaboration scenarios such as cooperation, non-cooperation, leader-follower, etc. can be simulated. The interface templates are presented by the authors using an example of design and manufacturing of a separation channel for a microscale gas chromatography system. Readers interested in the example are referred to Schnell et al. (2006).

Figure 17 Linking decision templates via interaction templates



By separating the interfaces from the formulation of decisions, the individual domain specific decisions can be standardised (see Requirement 3 in Table 1) while still maintaining the flexibility in configuration of the web-based virtual enterprises (see Requirement 2 in Table 1). The flexibility provided by the decision and interaction templates facilitates the enterprises to change the design processes quickly, thereby providing agility. Although, the concepts are presented in the context of design decisions only, they can be adapted for other types of decisions in an enterprise.

5 Conclusion

There is significant potential for applications of the decision-centric framework in the organisations characterised by numerous interactions among distributed stakeholders. The framework also facilitates strategic design and fast development of future product generations. Using the proposed framework, it will be possible to treat previously instantiated product development processes as reusable assets in maintaining the profitability of a product over its life cycle. The existing processes can be subsequently extended by means of derivative, variant and adaptive design.

The proposed decision-centric framework provides designers with a means to model their specific domains of application in terms of process critical decisions and any required information content so that their perspectives are easily communicated.

The framework will provide process (decision-makers acting primarily at the systems level) managers with the means to structure, organise and model the processes in which the designers are participating.

The fundamental goals are to

1 manage value chain processes
2 facilitate the collaboration of stakeholders and
3 effectively share information.

Processes are structured in terms of the decisions required for their resolution. The domain of application primarily discussed in this document is Distributed Collaborative Design and Manufacture. Specifically, we focus on industries with short product life cycles and high Time-to-Market (TTM) pressure, that rely on product family design in achieving mass customisation through leveraging common product platforms and modular components.

The future research tasks involve

1 modelling interactions between the value chain process elements and associated information flows
2 modelling stakeholder relationships commonly encountered throughout the value chain
3 capturing design process interactions using templates that can serve as a springboard for knowledge capture and
4 establishing communications protocols to represent the underlying interactions for enabling required information transfers.

Acknowledgements

We would like to acknowledge the help of Marco Fernandez and Andrew Schnell for providing their valuable insight into the decision-centric framework discussed in this paper.

References

Bascaran, E., Bannerot, R.B. and Mistree, F. (1989) 'Hierarchical selection decision support problems in conceptual design', *Engineering Optimization*, Vol. 14, pp.207–238.

Cochran, J.K., Lee, K.J., McDowell, D.L. and Sanders, T.H. (2000) 'Low density monolithic honeycombs by thermal chemical processing', *Proceedings of the 4th Conference on Aerospace Materials, Processes, and Environmental Technology*, Huntsville, AL.

Corbett, J., Meleka, J., Pym, C. and Dooner, M. (1991) *Design for Manufacture: Strategies, Principles and Techniques*, Addison-Wesley.

Fernández, M.G., Seepersad, C.C., Rosen, D., Allen, J.K. and Mistree, F. (2005) 'Decision support in concurrent engineering – the utility-based selection decision support problem', *Concurrent Engineering Research and Applications*, Vol. 13, No. 1, pp.13–28.

Goldman, S.L., Nagel, R.N. and Preiss, K. (1995) *Agile Competitors and Virtual Organizations: Strategies for Enriching the Customer*, Van Nostrand Reinhold.

Gu, P., Hashemian, M. and Nee, A.Y.C. (2004) 'Adaptable design', *Annals of CIRP*, Vol. 53, No. 2, pp.539–557.

Hayes, A.M., Wang, A., Dempsey, B.M. and McDowell, D.L. (2001) 'Mechanics of linear cellular alloys', *Proceedings of IMECE, International Mechanical Engineering Congress and Exposition*, New York, NY.

Hazelrigg, G.A. (1998) 'A framework for decision-based engineering design', *Journal of Mechanical Design*, Vol. 120, No. 4, pp.653–658.

Herrmann, J.W. and Schmidt, L.C. (2002) 'Viewing product development as a decision production system', *ASME 2002 Design Theory and Methodology Conference*, Montreal, Canada, Paper Number: DETC2002-DTM34030.

Jiang, Z. and Fung, R.Y.K. (2003) 'An adaptive agile manufacturing control infrastructure based on TOPNs-CS modelling', *International Journal of Advanced Manufacturing Technology*, Vol. 22, pp.191–215.

Kusar, J., Duhovnik, J., Grum, J. and Starbek, M. (2004) 'How to reduce new product development time', *Robotics and Computer-Integrated Manufacturing*, Vol. 20, pp.1–15.

Lau, H.C.W., Wong, C.W.Y. and Pun, K.F. (2003) 'Virtual agent modelling of an agile supply chain infrastructure', *Management Decision*, Vol. 41, No. 7, pp.625–634.

Maropoulos, P.G., Armoutis, N.D., Bramall, D.G., Lomas, C.D.W., Chapman, P. and Rogers, B. (2004) 'Agile design and manufacturing in collaborative networks for the defence industry', *IFAC Conference on Manufacturing, Modelling, Management and Control*, Athens, Greece.

Marston, M. (2000) 'Game based design: a game theory based approach to engineering design', PhD Dissertation, Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA.

Matthews, P.C., Lomas, C.D.W., Armoutis, N.D. and Maropoulos, P.G. (2006) 'Foundations of an agile design methodology', *International Journal of Agile Manufacturing*, Vol. 9, No. 1, pp.29–38.

Meyer, M.H. (1997) 'Revitalize your product lines through continuous platform renewal', *Research Technology Management*, Vol. 40, No. 2, pp.17–28.

Mistree, F., Hughes, O.F. and Bras, B.A. (1992) 'The compromise decision support problem and the adaptive linear programming algorithm', in M.P. Kamat (Ed). *Structural Optimization: Status and Promise*, Vol. 150, Washington, DC: AIAA, pp.251–290.

Mistree, F., Hughes, O.F. and Bras, B.A. (1993) 'The compromise decision support problem and the adaptive linear programming algorithm', in M.P. Kamat (Ed). *Structural Optimization: Status and Promise*, Washington, DC: AIAA, pp.247–286.

Mistree, F., Muster, D., Shupe, J.A. and Allen, J.K. (1989) 'A decision-based perspective for the design of methods for systems design', *Recent Experiences in Multidisciplinary Analysis and Optimization*, Hampton, Virginia, Paper Number: NASA CP 3031.

Muster, D. and Mistree, F. (1988) 'The decision support problem technique in engineering design', *International Journal of Applied Engineering Education*, Vol. 4, No. 1, pp.23–33.

Pahl, G. and Beitz, W. (1996) *Engineering Design – A Systematic Approach*, Berlin, Heidelberg, NY: Springer.

Panchal, J.H., Fernández, M.G., Allen, J.K., Paredis, C.J.J. and Mistree, F. (2005) 'Facilitating meta-design via separation of problem, product, and process information', *2005 ASME International Mechanical Engineering Congress and Exposition*, Orlando, FL, Paper Number: IMECE2005-80013.

Panchal, J.H., Fernández, M.G., Paredis, C.J.J. and Mistree, F. (2004) 'Reusable design processes via modular, executable, decision-centric templates', *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, NY, Paper Number: AIAA-2004-4601.

Peak, R., Lubell, J., Srinivasan, V. and Waterbury, S. (2004) 'STEP, XML, and UML: complementary technologies', *Journal of Computing and Information Science in Engineering: Special Issue on Engineering Information Management to Product Lifecycle Management*, Vol. 4, No. 4, pp.379–390.

Schnell, A.R., Messer, M., Panchal, J.H., Allen, J.K., Hesketh, P.J. and Mistree, F. (2006) 'Extending reusable decision-centric templates to facilitate collaboration using game theory', *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Portsmouth, Virginia, Paper Number: AIAA-MDO-63412.

Seepersad, C.C., Cowan, F.S., Chamberlain, M.K. and Mistree, F. (2002) 'Strategic design: leveraging and innovation for a changing marketplace', *Engineering Design Conference*, King's College, London, pp.3–20.

Simon, H.A. (1996) *The Sciences of the Artificial*, Cambridge, MA: The MIT Press.

Suh, N.P. (1990) *Principles of Design,* Oxford University Press, Oxford, UK**.**

Simpson, T., Lautenschlager, U. and Mistree, F. (1998) 'Mass customization in the age of information: the case for open engineering systems', W.H.R.A.L. Porter (Ed). *The Information Revolution Current and Future Consequences*, Greenwhich, CT: Ablex Publishing Corporation, pp.49–74.

StreamlineSCM (2004) *Enterprise Transaction Model*, Available at: http://www.streamlinescm.com/.

Supply Chain Council (2004) *Supply Chain Operations Reference (SCOR) Model*, Available at: http://www.supply-chain.org/slides/SCOR5.0OverviewBooklet.pdf.

Thurston, D.L. (1999) 'Real and perceived limitations to decision based design', *ASME DETC, Design Theory and Methodology*, Las Vegas, NV, Paper Number: DETC99/DTM-8750.

Xiao, A., Zeng, S., Allen, J.K., Rosen, D.W. and Mistree, F. (2002) 'Collaborating multi-disciplinary decision-making using game theory and design capability indices', *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, GA, Paper Number: AIAA-2002-5617.